

MEMORY REDUNDANCE SYSTEM

Publication number: JP53000032

Publication date: 1978-01-05

Inventor: TAKEZONO TAKASHI

Applicant: FUJITSU LTD

Classification:

- international: G06F12/16; G06F11/00; G06F11/16; G11C29/00;
G11C29/04; G06F12/16; G06F11/00; G06F11/16;
G11C29/00; G11C29/04; (IPC1-7): G06F11/00;
G11C9/00; G11C29/00

- european:

Application number: JP19760074876 19760623

Priority number(s): JP19760074876 19760623

Report a data error here

Abstract of **JP53000032**

PURPOSE: To secure a selection by switching a defective module to an optional good-quality module without specifying the position for the auxiliary module.

Data supplied from the **esp@cenet** database - Worldwide

①日本国特許庁

①特許出願公開

公開特許公報

昭53—32

⑤Int. Cl. ²	識別記号	⑥日本分類	庁内整理番号	④公開	昭和53年(1978)1月5日
G 11 C 29/00		97(7) C 01	7056—56	発明の数	1
G 06 F 11/00		97(7) C 02	6453—56	審査請求	未請求
G 11 C 9/00		97(7) G 12	6676—56		

(全 7 頁)

④記憶装置冗長方式

川崎市中原区上小田中1015番地
富士通株式会社内

①特 願 昭51—74876

①出 願 人 富士通株式会社

②出 願 昭51(1976)6月23日

川崎市中原区上小田中1015番地

⑦発 明 者 竹園隆

⑦代 理 人 弁理士 玉蟲久五郎 外 3 名

明 細 書

1. 発明の名称

記憶装置冗長方式

2. 特許請求の範囲

1, 0 を要素とするガロア体 $GF(2)$ の上の m 次の既約多項式を法とし、該多項式の根の冪乗を要素とするガロア体 $GF(2^m)$ のうち、要素の位数が 2^{m+i} であるガロア体を使用し、 m ビットのアドレス情報により決定される 2^m 個のアドレスのそれぞれにガロア体の 2^{m+i} 個の要素のうちの 2^m 個の要素を対応づけ、一方、チップ上には 2^{m+i} 個のモジュールを用意し、該モジュールのそれぞれにガロア体の 2^{m+i} 個の要素をアドレス・コードとして割り付け、該 2^m 個のアドレスと 2^m 個のガロア体の要素との対応づけを変えることにより、 2^{m+i} 個のモジュール中に i 個までのバースト不良モジュールの存在を許容することを特徴とする記憶装置冗長方式。

3. 発明の詳細な説明

本発明は、記憶装置冗長方式、特にメモリ・モ

ジュールに一部不良ビットが含まれていても、良品モジュールのみをアクセスするようにした冗長方式に関するものである。

最近の半導体技術では、素子の高密度化を図るためにウエハ・メモリの実現が検討されている。このウエハ・メモリを実現する場合、重要な問題は製造時の歩留りである。歩留りを向上させるためには、プロセス技術的、回路技術的に多くの困難を克服しなければならないが、あらかじめ冗長性を持たせた設計を行なっておけば、この困難を軽減することができる。また、信頼性の観点からも、冗長性を持たせた設計を行うことは、きわめて重要である。

冗長性を持たせる手段としては、ECCに代表されるソフト・ウエア的手法と、予備モジュールを用意しておき、不良メモリ・モジュールをこれに切換えて使用するような方法に代表されるハード・ウエア的手法とがある。

このうち、後者の方式においては、例えば、特開昭47-7060号公報、特開昭48-16536号公報等

に示されるように、予備モジュールの領域あるいはモジュール内の予備チップの領域があらかじめ定められており、不良モジュールあるいは不良チップがアクセスされたことを識別すると、特定の予備モジュール、予備チップに切換えている。

本発明は、従来の冗長方式のように予備モジュールの領域を限定することなく、モジュール全体として必要数に少くとも1個の冗長性を持たせ、不良モジュール以外の任意のモジュールを予備として用いることにより、不良モジュールをアクセスさせないようにすることを目的とするもので、良品モジュールのみを接続するために、ガロア体の要素をアドレス・コードとして割付けることを特徴としている。

先ず、第1図(a)に示すように、通常の3ビットの情報で選択される8個のモジュール群を考えてみる。

いま、外部より与えられるアドレス情報が(000)~(011)までの7通りであり、(111)のアドレスは与えられないものと仮定する。すなわち、実際

に必要なモジュールの数は7個で、他の1個は予備である。第1図(a)で明らかのように、あるモジュールに着目すると、その前後のモジュールに割付けられたアドレス・コードは、常に着目したモジュールに割付けられたアドレス・コードより、1を減じたもの、および1を加えたものに等しい。したがって、当然、各モジュールにアドレス・コードとして割付けられている3ビットの組合せは、3ビットのカウンタによって順次発生される周期8の数列の各要素に等しくなっている。

さて、第1図(a)のモジュール群において、アドレス・コード(010)のモジュールが不良であったとする。

そこで、外部より与えられるアドレス情報 $ADRS$ を第1図(a)に示すように加算器 ADD に入力し、(110)を加算して、この結果を内部アドレス情報として使用することを考えてみる。このようにすると、外部から与えられないこととしたアドレス(111)は、内部では(010)に相当するようになり、このことは結果的に不良モジュールがアクセスされな

くなったことを意味する。

他の7個のモジュールに対する外部アドレス情報 $ADRS$ も、それぞれ第1図(a)に示すようにアドレス変換されて内部アドレス $adre$ となる。したがって、このモジュール群は、見掛上第1図(b)に示すように、不良モジュールの次のモジュールから順にアドレス・コードが割付けられたことになる。

さて、前述の例は、説明のために通常の3ビットのバイナリ情報を使用し、(111)がアクセスされないという条件をつけたが、実際はこのような条件が成立する場合は殆んどなく、この例は実用的ではない。

そこで、次にバイナリ情報を使用せず、別の数列群の要素をアドレス・コードとして各モジュールに割付けることを考える。

すなわち、 $2^n + 1$ 個のモジュール群から 2^n 個のモジュールのみを選択できるようなアドレス・コードの割付け方法を考えてみる。

さて、代数学の教示するところによれば、ガロ

ア体 $GF(2^n)$ は、1, 0 をその要素とするガロア体 $GF(2)$ の上の n 次の既約多項式を法とする多項式環の剰余類からなっている。したがって、いま既約多項式 $F(x)$ の根を α とすると、 $GF(2^n)$ の諸要素は、 $1, \alpha, \alpha^2, \dots, \alpha^{n-1}$ の線形結合で下記のように表わすことができる。

$$a_0 + a_1 \alpha + a_2 \alpha^2 + \dots + a_{n-1} \alpha^{n-1}$$

また、別の表わし方として、係数 $a_0, a_1, a_2, \dots, a_{n-1}$ だけをとったベクトルでも表わせる。

$$(a_0, a_1, a_2, \dots, a_{n-1})$$

さらに、また $GF(2^n)$ から零の要素を除いた残りの要素は巡回群をなし、一般に $GF(2^n)$ の各要素は α の冪乗で表わせる。

いま、一例として $GF(2)$ 上の3次の既約多項式 $F(x) = x^3 + x + 1$ を法とする多項式環であるガロア体 $GF(2^3)$ を考えてみると、 $GF(2^3)$ のそれぞれの要素は第2図のように表わすことができる。

すなわち、 $F(x) = x^3 + x + 1$ より、 $x^3 + x + 1 = 0$ の根は $1, \alpha, \alpha^2$ の線形結合で表わされるから、正負の符号を考慮せずに、 $1 + \alpha + \alpha^2 = 0$ の α^3 の項を右

辺に移項すれば、 $1+\alpha=\alpha^2$ となって、 $1+\alpha$ は α^2 で乗算表示される。同じようにして、 $1+\alpha=\alpha^2$ の両辺に α を乗算すれば、 $\alpha+\alpha^2=\alpha^2\cdot\alpha$ となるから、 $\alpha+\alpha^2$ は α^4 で表わされ、 $1+\alpha=\alpha^2$ の両辺に α^2 を加えれば、 $1+\alpha+\alpha^2=\alpha^2+\alpha^2=\alpha^2(\alpha+1)=\alpha^6$ であるから、 $1+\alpha+\alpha^2$ は α^6 で表わされる。

つまり、この $GF(2^3)$ 上の各要素は周期 2^3-1 の巡回数列であり、ある要素に着目した場合、その前後の要素はその着目した要素を α で除したものの、および α を乗じたものに等しい。

したがって、 $GF(2^n)$ の中で、その要素の位数が 2^m+1 となるものを見出して、そのガロア体の法多項式を使用し、かつモジュールの各アドレス・コードとしてこのガロア体の要素を割付け、演算処理としては、第1図(a)の加算回路のかわりに α 乗算回路を使用すれば、最初の例(第1図(a)(b)(c))の非実用性は改善される。

以下、実施例について説明する。

いま、その要素の位数が 2^3+1 であるガロア体を探すと、 $GF(2^3)$ の中にあり、この法多項式は

ール・アドレス・レジスタ REG に入力される。そして、もし ECC 回路でエラーの検出が行われ、このモジュールに関するエラー履歴より CPU によってモジュール切換の必要があると判断されると、 SET 信号とシンδροーム(ポジション情報 POS) が第4図のデコーダ回路 DEC に入力される。これにより、指定されたビット位置(列)の不良モジュール・アドレス・レジスタ REG に SET 信号が入力され、このときのアドレスがラッチされる。このとき、同時に SET 信号もラッチされて、この列で切換えが行われたことも記憶する。

3ビットのバイナリ情報から6ビットの $GF(2^3)$ の要素に変換されたアドレス情報は、各モジュール列ごとに設けられた α 乗算回路($\times\alpha^k$)に入力される。 k の値は、アドレス・レジスタ REG にラッチされた情報によって制御される。この真理値表を第6図に示す。

第7図は、第4図における α 乗算回路の詳細接続図である。なお、第6図における「選択入力」は、第7図中のマルチプレクサ MPX のいずれの入

$P(z)=1+z^3+z^6$ である。また、このガロア体の各要素は第3図に示すとおりである。

第4図は本発明の一実施例を示す記憶装置の構成図である。第4図において、各モジュールは $4K \times 16$ bit で構成され、このモジュールが 9×22 のマトリクスに配置されて、全体として $32K \times 22$ bit の $SEC-DED$ メモリ・システムを構成している。なお、図中、 $4K$ 選択用アドレス 12 ビットおよび ECC 関連回路は、本発明に直接関係がないため省略してある。

各モジュール選択用コードとして、上記 $GF(2^3)$ の各要素が割付けられ、例えば、(100000)の割付けられたモジュールは、アドレス信号の6ビットが(100000)となったときにイネーブルされる。

外部より入力される8個のモジュールを選択するためのアドレス信号 $ADRS$ 3ビットは、先ず、読出専用メモリ(以下 ROM)に入力され、6ビットの $GF(2^3)$ の要素(α^k)に変換される。この ROM の内容を第5図に示す。また同時に、このアドレス信号 $ADRS$ は、各列ごとに設けられた不良モジュ

力端子の情報が出力されるかを示している。

α 乗算回路($\times\alpha^k$)は、次の方法で設計される。いま、 $GF(2^3)$ 中の任意のエレメントを $(a_0, a_1, a_2, a_3, a_4, a_5)$ とし、これを多項式の形で表わすと、次式が成立する。

$$\begin{aligned} \alpha \times (a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4 + a_5 z^5) \\ = b_0 + b_1 z + b_2 z^2 + b_3 z^3 + b_4 z^4 + b_5 z^5 \\ \text{mod } z^6 + z^3 + 1 \end{aligned}$$

これを解いて、 a_0, \dots, a_5 と b_0, \dots, b_5 の関係を求めると以下のようになる。

$$\begin{aligned} b_0 &= a_5, & b_1 &= a_0, & b_2 &= a_1, \\ b_3 &= a_2 + a_5, & b_4 &= a_3, & b_5 &= a_4 \end{aligned}$$

同じようにして、 $\alpha^2, \alpha^3, \dots$ を算出すると、第8図に示すようになる。第4図は、第8図に基づいて論理を構成したものである。

第4図において、例えば最初の列の3番目のモジュール(001000)が不良であった場合、 $GF(2^3)$ の要素(001000)に対応する外部アドレス情報 $ADRS$ は第5図に示すように(010)であるので、この列のアドレス・レジスタ REG には、第6図の

ように(1010)がセットされる。この結果、 α 乗算回路($\times \alpha^k$)の k の値は3になる。そして、この列に対して分配されるアドレス信号は第9図に示すようになる。第9図より明らかなように、不良モジュール(001000)の次のモジュールから順に、変換後のアドレス・コードが割付けられるため、不良モジュールの前のモジュールまでで 2^3 個が選択される。したがって、不良モジュール(001000)は外部よりアクセスされなくなり、かつ良品モジュール8個がアクセス可能となって、切換動作が行われたことになる。

実施例では、必要個数に1個だけ冗長性を持たせた場合を説明したが、任意の個数だけ予備モジュールを用意することにより、予備モジュールと同一個数のバースト不良モジュールを許容することができる。例えば、チップ上に 2^{m+i} 個のモジュールを用意し、各々にガロア体 $GF(2)$ の 2^{m+i} 個の要素をアドレス・コードとして割付けておき、連続した i 個までの不良モジュールが検出されたときには、アドレスとガロア体の要素との対応づ

けを変化させて、不良モジュールの次のモジュールから順に変換後のアドレス・コードを割付けることにより、良品モジュール 2^m 個のみをアクセスさせることが可能となる。

このように、本発明によれば、予備モジュールの位置を特定することなく、不良モジュールを任意の良品モジュールに切換えて選択させることができ、しかも、ガロア体の要素をアドレス・コードとして割付けることにより、外部より与えられるアドレス情報に何ら制限を設けなくてよいから、きわめて実用的であり、かつ簡単な構成により、製造時の歩留りを向上させることができる。

4. 図面の簡単な説明

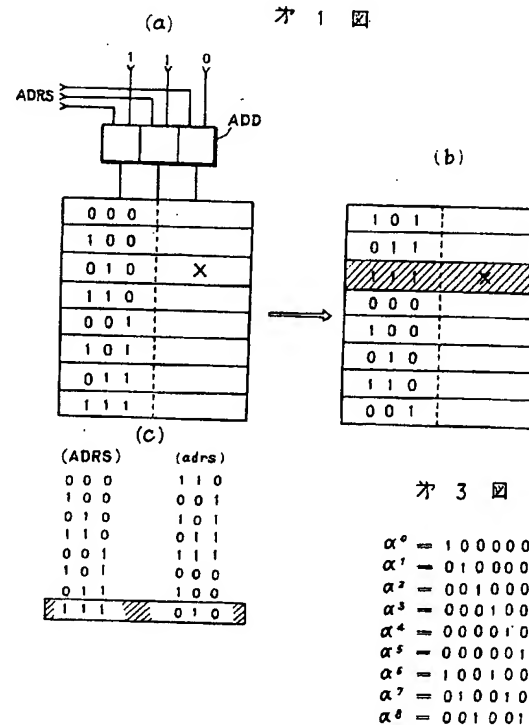
第1図は本発明の原理説明図、第2図はガロア体 $GF(2)$ 上の3次の既約多項式 $F(x) = x^3 + x + 1$ を法とする $GF(2^3)$ の要素の表示方法説明図、第3図は6次の既約多項式 $F(x) = x^6 + x^5 + 1$ を法とする $GF(2^6)$ の要素の説明図、第4図は本発明の一実施例を示す記憶装置のブロック構成図、第5図は第4図における読取専用メモリ(ROM)の記憶内容説明

図、第6図は α 乗算回路の情報説明図、第7図は第4図における α 乗算回路($\times \alpha^k$)の詳細接続図、第8図は第4図の α 乗算回路を設計するためのエレメントの関係図、第9図は第4図における変換アドレスの一例を示す説明図である。

ADRS: 外部よりのモジュール選択アドレス、
ADD: 加算器、ROM: 読取専用メモリ、SET: セット情報、DEC: デコーダ回路、POS: ポジション情報、REG: 不良モジュール・アドレス・レジスタ、 $\times \alpha^k$: α 乗算回路、I/O BUF: 入出力バッファ、SELSIG: モジュール選択信号、EOR: エクス・クルーシブ・オア回路、SP: 切換情報、MPX: マルチプレクサ、

特許出願人 富士通株式会社

代理人弁理士 玉 島 久 五 郎 (外3名)



オ 2 図

線形結合	ベクトル	α の冪乗
0	(000)	
1	(100)	$\alpha^0 = \alpha^7 = \alpha^{14}$
α	(010)	$\alpha^1 = \alpha^6 = \alpha^{15}$
α^2	(001)	$\alpha^2 = \alpha^5 = \dots$
$1 + \alpha$	(110)	$\alpha^3 = \alpha^{10} \dots$
$\alpha + \alpha^2$	(011)	$\alpha^4 = \alpha^{11} \dots$
$1 + \alpha + \alpha^2$	(111)	$\alpha^5 = \alpha^{12}$
$1 + \alpha^2$	(101)	$\alpha^6 = \alpha^{13}$

オ 5 図

ROMアドレス	ROM 情報
0 0 0	1 0 0 0 0 0 — α^0
1 0 0	0 1 0 0 0 0 — α^1
0 1 0	0 0 1 0 0 0 — α^2
1 1 0	0 0 0 1 0 0 — α^3
0 0 1	0 0 0 0 1 0 — α^4
1 0 1	0 0 0 0 0 1 — α^5
0 1 1	1 0 0 1 0 0 — α^6
1 1 1	0 1 0 0 1 0 — α^7
$A_0 A_1 A_2$	$d_0 d_1 d_2 d_3 d_4 d_5$

オ 4 図

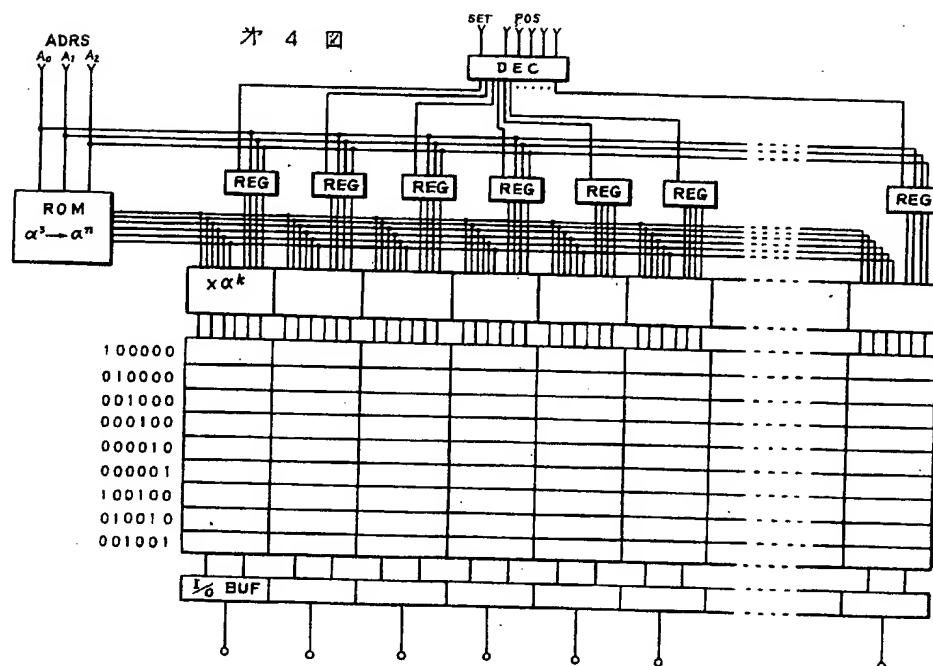


図 6

アドレスレジスタ REG SET				α^k	選択入力
A ₀	A ₁	A ₂	A ₃		
0	0	0	0	α^0	1
1	0	0	0	α^1	2
1	1	0	0	α^2	3
1	0	1	0	α^3	4
1	1	1	0	α^4	5
1	0	0	1	α^5	6
1	1	0	1	α^6	7
1	0	1	1	α^7	8
1	1	1	1	α^8	9

図 8

	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8
b ₀	a ₅	a ₄	a ₃	a ₂ +a ₅	a ₁ +a ₄	a ₀ +a ₃	a ₂	a ₁
b ₁	a ₀	a ₅	a ₄	a ₃	a ₂ +a ₅	a ₁ +a ₄	a ₀ +a ₃	a ₂
b ₂	a ₁	a ₀	a ₅	a ₄	a ₃	a ₂ +a ₅	a ₁ +a ₄	a ₀ +a ₃
b ₃	a ₂ +a ₅	a ₁ +a ₄	a ₀ +a ₃	a ₂	a ₁	a ₀	a ₅	a ₄
b ₄	a ₃	a ₂ +a ₅	a ₁ +a ₄	a ₀ +a ₃	a ₂	a ₁	a ₀	a ₅
b ₅	a ₄	a ₃	a ₂ +a ₅	a ₁ +a ₄	a ₀ +a ₃	a ₂	a ₁	a ₀

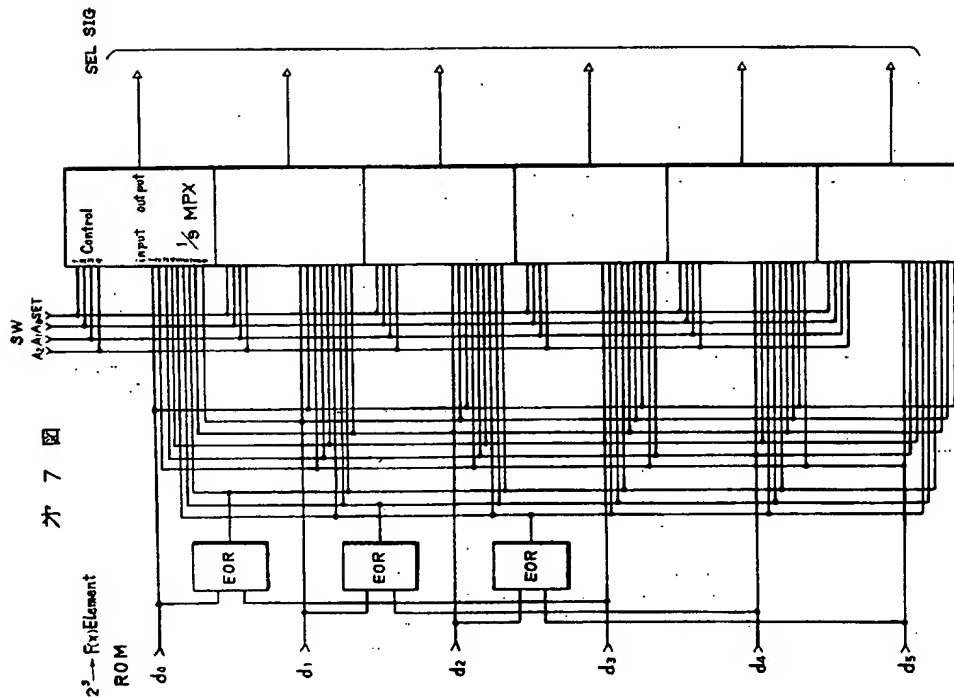


図 7

オ 9 図

外部アドレス情報	変換前内部アドレス	変換後アドレス情報 ($\times \alpha^2$)
0 0 0	1 0 0 0 0 0 (α^0)	0 0 0 1 0 0 α^2
1 0 0	0 1 0 0 0 0 (α^1)	0 0 0 0 1 0 α^4
0 1 0	0 0 1 0 0 0 (α^2)	0 0 0 0 0 1 α^5
1 1 0	0 0 0 1 0 0 (α^3)	1 0 0 1 0 0 α^6
0 0 1	0 0 0 0 1 0 (α^4)	0 1 0 0 1 0 α^7
1 0 1	0 0 0 0 0 1 (α^5)	0 0 1 0 0 1 α^8
0 1 1	1 0 0 1 0 0 (α^6)	1 0 0 0 0 0 $\alpha^9 = \alpha^0$
1 1 1	0 1 0 0 1 0 (α^7)	0 1 0 0 0 0 $\alpha^{10} = \alpha^1$